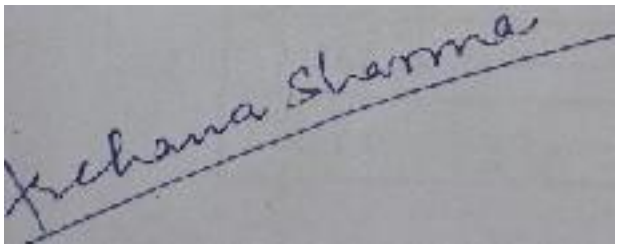
Department of MCA

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Computer Architecture & VLSI Design
2.	Course Code	MCA-103
3.	Academic Year	2024-25
4.	Semester	1ST
5.	Number of Lesson plans	37
6.	Faculty Assigned	Dr.Archana Sharma



Faculty Signature

Lesson Plan No. 1	**Course Name:** Computer Architecture & VLSI Design **Topic:** Von Neumann Architecture	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic components of a computer system. b. Learn the concept of the stored program concept. c. Recognize the role of the control unit, ALU, memory, and I/O devices in computer operation.
Teaching Aids (if any)	a. Interactive Whiteboards b. Charts and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question if they have ever seen a computer or smartphone. Explain that these devices are made up of many components that work together to process information. Introduce the concept of computer architecture, which defines the structure and organization of a computer system. 2. **Development** (30 minutes) **a. Stored Program Concept:** Explain that the stored program concept is a fundamental principle of modern computers. Discuss how instructions and data are stored in the same memory unit. Highlight the advantages of this approach, such as flexibility and programmability. **b. Components of the Von Neumann Architecture:** **Control Unit:** Describe its role in fetching, decoding, and executing instructions. Explain how it coordinates the activities of other components. **Arithmetic Logic Unit (ALU):** Discuss its function in performing arithmetic and logical operations. Mention examples of operations like addition, subtraction, and comparison. **Memory:** Explain the difference between primary and secondary memory. Discuss the concept of addressing and how data is stored and retrieved. **Input/Output (I/O) Devices:** Introduce the various types of I/O devices (e.g., keyboard, mouse, monitor, printer).

	Explain how they interact with the computer system. 1. Exercise (5 minutes) – - What is the stored program concept? - Name the four main components of the Von Neumann architecture. - What is the function of the control unit? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - https://en.wikipedia.org/wiki/Von_Neumann_architecture - https://en.wikipedia.org/wiki/Computer_architecture Homework - Identify and discuss examples of how the principles of Von Neumann architecture are applied in modern computing systems. and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Nearpod Quiz on Von Neumann architecture . Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	**Course Name:** Computer Architecture & VLSI Design **Topic:** Digital and Analog Systems	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental differences between digital and analog signals. b. Learn about the advantages and disadvantages of digital and analog systems. c. Explore real-world applications of digital and analog technologies.
Teaching Aids (if any)	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. if they have ever heard of "digital" or "analog" terms. Explain that these terms refer to different ways of representing information. - Use simple examples like comparing a traditional clock (analog) to a digital watch. - Introduce the concept of digital" or "analog" . Show Figure on slide. 2. **Development** (30 minutes) **a. Digital Systems:** Define digital signals as discrete values (e.g., 0s and 1s). Explain the concept of binary code. Discuss the advantages of digital systems, such as noise immunity, accuracy, and ease of processing. **b. Analog Systems:** Define analog signals as continuous values. Explain how analog signals can represent a wide range of information (e.g., sound, images). Discuss the advantages of analog systems, such as naturalness and human perception. c. **Comparing Digital and Analog:** Compare and contrast digital and analog systems based on their characteristics. Discuss the process of analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC). Explore real-world examples of digital and analog technologies (e.g., smartphones, radios, TVs). 3. Exercise (5 minutes) – - What is the main difference between digital and analog signals? - Name one advantage of digital systems over analog systems.

	- What is the process of converting an analog signal into a digital signal called? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - **Digital and Analog Signals:** https://m.youtube.com/watch?v=COE9wjcdPhQ - **Analog to Digital Conversion:** https://www.electronics-tutorials.ws/ - **Digital vs. Analog: A Simple Explanation:** https://www.youtube.com/watch?v=btgAUdbj85E Homework - Provide examples of real-world applications where analog systems are preferred over digital systems, and vice versa and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Nearpod Quiz on Digital and Analog Systems Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	**Course Name:** Computer Architecture & VLSI Design **Topic:** Number Systems, Their Types, and Conversions	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of different number systems (decimal, binary, octal, hexadecimal). b. Learn how to convert numbers between different number systems. c. Apply the knowledge of number systems in real-world applications.
Teaching Aids (if any)	a. Interactive Whiteboards b. Educational Software/Apps
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. importance of number systems in computer science and other fields. what a number system is and provide examples of commonly used ones. Introduce the concept of place value in different number systems. 2. **Development** (30 minutes) **a. Decimal Number System:** Explain the concept of place value in the decimal system. Demonstrate how to convert decimal numbers to other number systems. **b. Binary Number System:** Explain the concept of place value in the binary system. Demonstrate how to convert binary numbers to other number systems and vice versa. **c. Octal and Hexadecimal Number Systems:** Explain the concept of place value in the octal and hexadecimal systems. Demonstrate how to convert numbers between these systems and the decimal system. 3. Exercise (5 minutes) – - Convert the decimal number 125 to binary. - What is the decimal equivalent of the binary number 1011? - Convert the hexadecimal number A2 to decimal. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - **Decimal to binary:** https://www.rapidtables.com/convert/number/decimal-to-

	binary.html?x= https://www.rapidtables.com/convert/number/binary-to-decimal.html?x=11111111111101110000 https://www.rapidtables.com/convert/number/hex-to-decimal.html?x=10 Homework - Discuss the significance of different number systems (binary, octal, decimal, hexadecimal) in digital electronics and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Nearpod Quiz on Number Systems, Their Types, and Conversions. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	**Course Name:** Computer Architecture & VLSI Design **Topic:** Binary arithmetic operations	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of binary numbers and their relationship to decimal numbers. b. Perform basic arithmetic operations (addition, subtraction, multiplication, and division) using binary numbers. c. Apply binary arithmetic in real-world applications.
Teaching Aids (if any)	a. Interactive Whiteboards b. Educational Software/Apps
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have heard of binary code. Explain that binary code is a system of representing numbers using only two digits: 0 and 1. Relate binary code to the on/off states of computers and other digital devices. 2. **Development** (30 minutes) **a. Binary Number System:** Explain the place value system in binary numbers. Demonstrate how to convert decimal numbers to binary and vice versa. Provide examples and practice problems. **b. Binary Addition:** Explain the rules for binary addition. Demonstrate binary addition using examples. Discuss the concept of carrying over in binary addition. **c. Binary Subtraction:** Explain the rules for binary subtraction. Demonstrate binary subtraction using examples. Discuss the concept of borrowing in binary subtraction. 3. Exercise (5 minutes) – - Convert the decimal number 15 to binary. - Perform the binary addition: 101 + 110. - Subtract 10 from 1001 in binary. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - **Binary arithmetic:** https://en.wikipedia.org/wiki/Category:Binary_arithmetic

	https://www.rapidtables.com/ Homework - Explain the borrowing process in binary subtraction. Perform the subtraction of $10010_2 - 1101_2$ and show all steps, including borrows. and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Nearpod Quiz on Binary Arithmetic Operations. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	**Course Name:** Computer Architecture & VLSI Design **Topic:** Representation of Negative Numbers: 1's Complement and 2's Complement	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of signed number representation. b. Learn the 1's complement and 2's complement methods for representing negative numbers. c. Perform basic arithmetic operations using 1's and 2's complement.
Teaching Aids (if any)	a. Interactive Whiteboards b. diagrams and examples
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever seen a negative number on a calculator or computer screen. - Explain that computers use specific methods to represent negative numbers in binary form. - Introduce the concept of signed number representation, which allows computers to distinguish between positive and negative values. 2. **Development** (30 minutes) **a. Signed Magnitude Representation:** Briefly discuss the signed magnitude method, where a separate sign bit is used to indicate the sign of a number. Explain the limitations of this method, such as the existence of two representations for zero. **b. 1's Complement:** Define 1's complement as a method where the negative of a number is obtained by inverting all its bits. Explain how to convert a positive number to its 1's complement and vice versa. Discuss the issue of two representations for zero in 1's complement. **c. 2's Complement:** Define 2's complement as a method where the negative of a number is obtained by inverting all its bits and adding 1. Explain how to convert a positive number to its 2's complement and vice versa. Discuss the advantages of 2's complement, such as having only one representation for zero and being suitable for arithmetic operations. 3. Exercise (5 minutes) – - What is the difference between signed magnitude and 1's complement representation?

	- How is the 2's complement of a number calculated? - What is the advantage of using 2's complement over 1's complement for arithmetic operations? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **https://en.wikipedia.org/wiki/Signed_number_representation** Homework Given the 4-bit 2's complement representation of -7, find its binary representation. Then, perform the addition of -7 and 3 using 2's complement and show the steps involved. and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	**Course Name:** Computer Architecture & VLSI Design **Topic:** Code Representation: BCD Code & Excess-3	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of code representation in digital systems. b. Learn the BCD code and its applications. c. Understand the Excess-3 code and its advantages..
Teaching Aids (if any)	a. Interactive Whiteboards b. examples
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever seen a digital clock or a calculator. - Explain that these devices use codes to represent numbers. - Introduce the concept of code representation, which is the way numbers and characters are represented using binary digits. 2. **Development** (30 minutes) **a. BCD Code:** Explain that BCD (Binary Coded Decimal) is a code that represents each decimal digit with a 4-bit binary code. Discuss the advantages of BCD, such as compatibility with decimal systems and ease of conversion. Provide examples of BCD representation for different decimal numbers. **b. Excess-3 Code:** Explain that Excess-3 code is a self-complementing code where each decimal digit is represented by adding 3 to its corresponding BCD code. Discuss the advantages of Excess-3 code, such as ease of subtraction and detection of errors. Provide examples of Excess-3 representation for different decimal numbers. **c. Comparison of BCD and Excess-3:** Compare and contrast BCD and Excess-3 codes based on their characteristics and applications. Discuss the trade-offs between the two codes in terms of ease of use, efficiency, and error detection. 3. Exercise (5 minutes) – What is BCD code? How does Excess-3 code differ from BCD code? What are the advantages of using Excess-3 code? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation

	from students on these. 2. Suggested Reading https://en.wikipedia.org/wiki/BCD_code https://en.wikipedia.org/wiki/Excess-3_code Homework Convert the decimal number 45 into both BCD and Excess-3 codes and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	**Course Name:** Computer Architecture & VLSI Design **Topic:** Logic Gates: AND, OR, NOT, NAND, XOR, NOR, XNOR	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of digital logic. b. Learn the truth tables and functions of basic logic gates (AND, OR, NOT). c. Explore the derived logic gates (NAND, XOR, NOR, XNOR) and their applications.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and animations
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever seen a computer or smartphone. - Explain that these devices use digital signals (0s and 1s) to process information. - Introduce the concept of logic gates, which are fundamental building blocks of digital circuits. 2. **Development** (30 minutes) **a. Basic Logic Gates:** **AND Gate:** Explain its truth table and function. Demonstrate its use in digital circuits. **b. OR Gate:** Explain its truth table and function. Demonstrate its use in digital circuits. **c. NOT Gate:** Explain its truth table and function. Demonstrate its use in digital circuits. **d. Derived Logic Gates:** **NAND Gate:** Explain its relationship to the AND gate. Demonstrate its use in digital circuits. **e. XOR Gate:** Explain its truth table and function. Demonstrate its use in digital circuits. **f. NOR Gate:** Explain its relationship to the OR gate. Demonstrate its use in digital circuits. **g. XNOR Gate:** Explain its relationship to the XOR gate. Demonstrate its use in digital circuits.

	1. Exercise (5 minutes) – What is the output of an AND gate when both inputs are 1? What is the function of a NOT gate? How is a NAND gate related to an AND gate? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **https://en.wikipedia.org/wiki/Logic_gate** Homework a. Create a truth table for the circuit showing all possible combinations of inputs A, B, and C, and the corresponding outputs of the AND, OR, and NOT gates. b. If the output of the AND gate is used as the input to a second NOT gate, what is the final output of the circuit? and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	**Course Name:** Computer Architecture & VLSI Design **Topic:** Boolean Laws and Their Expressions	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental laws of Boolean algebra. b. Learn how to simplify Boolean expressions using these laws. c. Apply Boolean algebra to solve logical problems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and animations
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Begin by asking students if they have ever encountered logical statements or decisions in their daily lives. - Explain that Boolean algebra provides a mathematical framework for representing and manipulating logical statements. - Introduce the concept of binary values (0 and 1) and their relationship to logical true and false. 2. **Development** (30 minutes) **a. Fundamental Boolean Laws:** **Identity Law:** Explain the identity law (A AND 1 = A, A OR 0 = A). Provide examples to illustrate its application. **Complement Law:** Discuss the complement law (A AND NOT A = 0, A OR NOT A = 1). Show how it is used to negate logical statements. **Commutative Law:** Introduce the commutative law (A AND B = B AND A, A OR B = B OR A). Demonstrate how the order of operands does not affect the result. **Associative Law:** Explain the associative law (A AND (B AND C) = (A AND B AND C, A OR (B OR C) = (A OR B) OR C). Show how grouping of operands does not change the outcome. **Distributive Law:** Discuss the distributive law (A AND (B OR C) = (A AND B) OR (A AND C), A OR (B AND C) = (A OR B) AND (A OR C)). Demonstrate how to distribute an operand over a logical operation. **b. Simplifying Boolean Expressions:** how to simplify Boolean expressions using the fundamental

	laws. examples of simplification problems and guide students through the process. 1. Exercise (5 minutes) – What is the identity law in Boolean algebra? How can you simplify the expression (A AND B) OR (A AND NOT B)? What is the complement of the expression A OR B? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **https://en.wikipedia.org/wiki/Logic_gate** Homework a. Create a truth table for the circuit showing all possible combinations of inputs A, B, and C, and the corresponding outputs of the AND, OR, and NOT gates. b. If the output of the AND gate is used as the input to a second NOT gate, what is the final output of the circuit? and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 9	**Course Name:** Computer Architecture & VLSI Design **Topic:** Representation in SOP, POS form and their simplifications	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. understand the concept of Sum of Products (SOP) and Product of Sums (POS) forms. b. learn how to represent Boolean expressions using SOP and POS forms. c. simplify Boolean expressions using Boolean algebra and Karnaugh maps.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and animations
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the concept of Boolean algebra and its use in digital logic design. - Introduce the two primary forms of representing Boolean expressions: SOP and POS. - Provide examples of simple Boolean expressions in SOP and POS forms. 2. **Development** (30 minutes) **a. SOP Form:** Define SOP form as a logical expression consisting of the logical OR (sum) of AND (product) terms. Explain how to convert a truth table into SOP form. Provide examples of converting truth tables into SOP form. **b. POS Form:** Define POS form as a logical expression consisting of the logical AND (product) of OR (sum) terms. Explain how to convert a truth table into POS form. Provide examples of converting truth tables into POS form. **c. Simplification:** Introduce Boolean algebra laws and theorems. Demonstrate how to simplify SOP and POS expressions using Boolean algebra. Explain the concept of Karnaugh maps and how to use them for simplification. Provide examples of simplifying expressions using Boolean algebra and Karnaugh maps. 1. Exercise (5 minutes) – Simplify the following SOP expression using Boolean algebra: F = AB' + AB + A'B

	Convert the following SOP expression into POS form: F = A'B + AB' + AB NOT B)? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://www.electronics-tutorials.ws/category/boolean **youTube Video:** https://m.youtube.com/watch?v=UfZKvPQku8w Homework a. Discuss the significance of SOP and POS forms in digital circuit design. b. Show the steps taken to simplify the POS expression. and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 10	**Course Name:** Computer Architecture & VLSI Design **Topic:** Code Converters	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. understand the concept of code conversion in digital electronics. b. learn about different types of code converters and their applications. c. identify the appropriate code converter for a given task.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and animations
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Begin by discussing the importance of representing data in different formats in digital systems. - Explain how code converters are used to transform data from one code to another. - Provide examples of common code conversion scenarios (e.g., BCD to binary, binary to Gray). **Development** (30 minutes) **a. Types of Code Converters:** Binary-to-BCD converter BCD-to-binary converter Binary-to-Gray converter Gray-to-binary converter Excess-3 to binary converter Binary to Excess-3 converter **b. Operation and Applications:** Explain the logic behind each type of code converter. Discuss the applications of code converters in various digital circuits (e.g., displays, microprocessors). Provide examples of how to design and implement code converters using logic gates. **c. Design Considerations:** Discuss the factors to consider when selecting a code converter for a specific application (e.g., speed, cost, complexity). Explore techniques for optimizing the design of code converters (e.g., using lookup tables). 1. Exercise (5 minutes) – - What is the purpose of a code converter? - How does a binary-to-BCD converter work? - Name three applications of code converters in digital systems. Use Nearpod to collect responses and discuss the answers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Digital Logic Design:** https://www.tutorialspoint.com/digital-electronics/index.htm **Code Converters:** https://www.geeksforgeeks.org/code-converters-binary-to-from-gray-code/ **Code Converters Tutorial:** https://www.youtube.com/watch?v=1Y5_ibL2u7Y Homework a. Discuss how code converters are utilized in data migration betweensystems. b. Explain the role of code converters in the context of IoT devices. and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 11	**Course Name:** Computer Architecture & VLSI Design **Topic:** Error Detection and Correction: Hamming Code	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of error detection and correction. b. Learn the Hamming code algorithm for error detection and correction. c. Apply the Hamming code to detect and correct single-bit errors in data transmission.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and animations
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the importance of reliable data transmission in digital communication. - Discuss the challenges posed by noise and errors in data transmission. - Introduce the concept of error detection and correction techniques. 2. **Development** (30 minutes) **a. Hamming Code:** Explain the concept of parity bits. Discuss the Hamming code algorithm for generating and checking parity bits. Demonstrate how the Hamming code can detect and correct single-bit errors. **b. Hamming Code Applications:** Highlight real-world applications of the Hamming code in various fields, such as computer memory, data storage, and communication systems. **c. Error Correction Capabilities:** Discuss the limitations of the Hamming code in detecting and correcting multiple-bit errors. Introduce other error correction techniques, such as Reed-Solomon codes, for handling more complex error scenarios**.** 1. Exercise (5 minutes) – - What is the purpose of parity bits in error detection and correction? - How does the Hamming code determine the location of a single-bit error? - Can the Hamming code correct multiple-bit errors? Explain. Use Nearpod to collect responses and discuss the answers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://en.wikipedia.org/wiki/Hamming_code https://www.geeksforgeeks.org/hamming-code-in-computer-network/ **YouTube Video Reference** https://www.youtube.com/watch?v=X8jsijhllIA Homework Discuss a real-world application of Hamming code in digital communication systems. Why is it important to detect and correct errors in such systems and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 12	**Course Name:** Computer Architecture & VLSI Design **Topic:** K Maps	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. understand the concept of K maps for simplifying Boolean expressions. b. learn how to construct K maps for different numbers of variables. c. apply K maps to minimize Boolean expressions and design digital circuits.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and animations
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly review Boolean algebra and its basic operations (AND, OR, NOT). - Explain the importance of simplifying Boolean expressions for efficient circuit design. - Introduce K maps as a graphical method for simplifying Boolean expressions.. 2. **Development** (30 minutes) **a. K Map Construction:** Explain the concept of a Karnaugh map (K map) and its grid structure. Demonstrate how to label the rows and columns of a K map based on the number of variables. Discuss the adjacency rules for grouping cells in a K map. **b. Grouping Cells and Minimization:** Show how to group adjacent cells containing 1s in a K map to form implicants. Explain the concept of essential prime implicants and their role in minimization. Demonstrate the process of minimizing a Boolean expression using K maps. **c. Examples and Applications:** Provide examples of K maps for different numbers of variables (2, 3, 4). Solve practice problems to reinforce the concepts of K map construction and minimization. Discuss real-world applications of K maps in digital circuit design. 1. Exercise (5 minutes) – - What is a K map used for? - How do you determine the size of a K map based on the number of variables?

	- What are the adjacency rules for grouping cells in a K map? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Boolean Algebra and K Maps:** **https://en.wikipedia.org/wiki/Karnaugh_map** **K Map Tutorial: https://www.youtube.com/watch?v=dx5Ixkt13Yk** **YouTube Video Reference:** **K Maps Tutorial:** **https://m.youtube.com/watch?v=FPrcIhqNPVo** Homework - Simplify the following expression considering the don't-care conditions:F(A, B, C) = Σ(0, 1, 2, 5) + d(3, 4) and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 13	**Course Name:** Computer Architecture & VLSI Design **Topic:** Code Converters	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of code conversion. b. Learn about different types of code converters. c. Analyze the applications of code converters in digital systems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the significance of code conversion in digital electronics. - Discuss the need for representing data in various codes. - Provide examples of common code systems (e.g., binary, BCD, Gray). 2. **Development** (30 minutes) **a. Types of Code Converters:** Binary-to-BCD converter BCD-to-binary converter Gray-to-binary converter Binary-to-Gray converter Excess-3-to-binary converter Excess-3-to-BCD converter **b. Operation of Code Converters:** Explain the logic behind the conversion process for each type. Use truth tables or Karnaugh maps to illustrate the conversion functions. **c. Applications of Code Converters:** Discuss real-world examples where code converters are used (e.g., digital displays, data transmission). Highlight the advantages of using code converters in specific applications. 1. Exercise (5 minutes) – - What is the purpose of a code converter? - How does a binary-to-BCD converter work? - Name three common types of code converters. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Code Converter Circuit Diagrams:** Digital Electronics - Code Converters

	Homework 3. Simplify the following expression considering the don't-care conditions:F(A, B, C) = Σ(0, 1, 2, 5) + d(3, 4) and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 14	**Course Name:** Computer Architecture & VLSI Design **Topic:** Error Detection and Correction: Hamming Code	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of error detection and correction in digital systems. b. Learn about the Hamming code and its principles. c. Apply Hamming code to detect and correct single-bit errors.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Discuss the importance of data integrity in digital communication and storage. - Explain the challenges posed by noise and interference in digital systems. - Introduce the concept of error detection and correction techniques. 2. **Development** (30 minutes) **a. Hamming Code Principles:** Explain the concept of parity bits. Discuss the Hamming distance and its significance. Derive the formula for calculating the number of parity bits required for a given data word length. **b. Hamming Code Encoding:** Demonstrate the process of adding parity bits to a data word to create a Hamming codeword. Use examples to illustrate the encoding procedure. **c. Hamming Code Decoding and Error Correction:** Explain how to detect errors by calculating parity bits. Describe the process of locating the error bit using the syndrome. Demonstrate the correction of single-bit errors. 1. Exercise (5 minutes) – - What is the purpose of error detection and correction in digital systems? - How does Hamming code work to detect and correct errors? - What is the syndrome in Hamming code, and how is it used? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading

	Hamming code: Error Correcting Codes - Hamming codes Error Correction and Detection Codes | CRC, Hamming, Parity **https://www.youtube.com/embed/1A_NcXxdoCc** Homework For the data word 0110, find the encoded (7,4) Hamming code, and demonstrate how to check for errors in the received code 0110110 and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 15	**Course Name:** Computer Architecture & VLSI Design **Topic:** Combinational and Sequential Circuits: Half & Full Adders & Subtractors	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concepts of combinational and sequential circuits. b. Learn the operations of half and full adders. c. Learn the operations of half and full subtractors. d. Analyze the applications of adders and subtractors in digital systems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the difference between combinational and sequential circuits. - Discuss the importance of addition and subtraction operations in digital electronics. - Provide examples of applications where adders and subtractors are used. 2. **Development** (30 minutes) **a. Combinational Circuits:** Define combinational circuits and their characteristics. Explain the concept of logic gates and their use in building combinational circuits. **b. Half Adder:** Describe the structure and operation of a half adder. Use truth tables to illustrate the addition of two single-bit numbers. **c. Full Adder:** Describe the structure and operation of a full adder. Use truth tables to illustrate the addition of three single-bit numbers. **d. Half Subtractor:** Describe the structure and operation of a half subtractor. Use truth tables to illustrate the subtraction of one single-bit number from another. **e. Full Subtractor:** Describe the structure and operation of a full subtractor. Use truth tables to illustrate the subtraction of one single-bit number from another with a borrow input. 1. Exercise (5 minutes) – - What is the difference between a combinational and a sequential circuit?

	- How does a half adder work? - What is the purpose of a borrow input in a full subtractor? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Hamming code:** **Half Adder in Digital Electronics** **https://www.youtube.com/embed/sUutDs7FFeA?** Homework Explain how you could use a full adder circuit to assist in converting binary numbers to decimal. Discuss the similarities and differences between adders and subtractors in terms of their logic design and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 16	**Course Name:** Computer Architecture & VLSI Design **Topic:** Parallel Adders	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of parallel addition. b. Learn the structure and operation of ripple-carry adders. c. Learn the structure and operation of carry-lookahead adders. d. Analyze the trade-offs between ripple-carry and carry-lookahead adders.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the need for efficient addition in digital systems. - Discuss the limitations of serial addition. - Introduce the concept of parallel addition. 2. **Development** (30 minutes) **a. Ripple-Carry Adders:** Describe the structure of a ripple-carry adder using full adders. Explain the propagation of carry signals through the adder. Analyze the delay associated with ripple-carry adders. **b. Carry-Lookahead Adders:** Explain the concept of carry-lookahead generation. Describe the structure of a carry-lookahead adder. Analyze the advantages of carry-lookahead adders in terms of speed. **c. Trade-offs Between Ripple-Carry and Carry-Lookahead Adders:** Discuss the factors that influence the choice between ripple-carry and carry-lookahead adders. Compare the area, speed, and power consumption of these adders. 1. Exercise (5 minutes) – - What is the difference between serial and parallel addition? - How does a ripple-carry adder work? - What is the advantage of a carry-lookahead adder over a ripple-carry adder? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Parallel Adder and Parallel Subtractor Parallel Adder and Parallel Subtractor - GeeksforGeeks

	https://www.youtube.com/embed/NO7Gt8IDSGA Homework Discuss some real-world applications where parallel adders are used. Why are they preferred in these applications? What factors influence the performance of a parallel adder? How can you optimize these factors in a design and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 17	**Course Name:** Computer Architecture & VLSI Design **Topic:** encoder, decoder	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concepts of encoders and decoders. b. Learn the different types of encoders and decoders. c. Analyze the applications of encoders and decoders in digital systems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the purpose of encoders and decoders in digital electronics. - Discuss the need for converting data between different formats. - Provide examples of common encoding and decoding schemes. 2. **Development** (30 minutes) **a. Encoders:** Define encoders and their function. Explain the operation of various types of encoders, such as: **b. Priority encoder** BCD-to-7-segment decoder Octal-to-binary encoder **c. Decoders:** Define decoders and their function. Explain the operation of various types of decoders, such as: **d. BCD-to-decimal decoder** 2-to-4 line decoder 3-to-8 line decoder **e. Applications of Encoders and Decoders:** Discuss real-world examples where encoders and decoders are used (e.g., keyboards, displays, digital circuits). Highlight the advantages of using encoders and decoders in specific applications. 1. Exercise (5 minutes) – - What is the difference between an encoder and a decoder? - How does a priority encoder work? - What is the purpose of a BCD-to-7-segment decoder? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Encoders and Decoders in Digital Logic - GeeksforGeeks Difference between Encoder and Decoder

	https://www.youtube.com/embed/NO7Gt8IDSGA Homework Design a 4-to-2 encoder with an enable input. How does the enable input affect the output and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 18	**Course Name:** Computer Architecture & VLSI Design **Topic:** Multiplexer and De-Multiplexer	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of multiplexing and demultiplexing. b. Learn the basic structure and operation of a multiplexer. c. Understand the basic structure and operation of a demultiplexer. d. Explore the applications of multiplexers and demultiplexers in digital systems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - the concept of data transmission and the need for efficient use of communication channels. - Explain how multiplexing allows multiple data streams to be transmitted over a single channel. - Introduce the concept of demultiplexing, which is the reverse process of selecting and extracting individual data streams from a multiplexed signal. 2. **Development** (30 minutes) **a. Multiplexer:** Define a multiplexer and its purpose. Explain the basic structure of a multiplexer, including the select lines, data inputs, and output. Discuss the operation of a multiplexer, how it selects one of the data inputs based on the select lines. Present examples of different types of multiplexers (e.g., 2-to-1, 4-to-1, 8-to-1). **b. Demultiplexer:** Define a demultiplexer and its purpose. Explain the basic structure of a demultiplexer, including the select lines, data input, and data outputs. Discuss the operation of a demultiplexer, how it directs the input data to one of the output lines based on the select lines. Present examples of different types of demultiplexers. **c. Applications:** Discuss various applications of multiplexers and demultiplexers in digital systems, such as: Communication systems Computer networks Digital signal processing Microprocessors and microcontrollers 1. Exercise (5 minutes) –

	- What is the purpose of a multiplexer? - How does a demultiplexer work? - Name two applications of multiplexers and demultiplexers. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Tutorials Point:** https://www.javatpoint.com/de-multiplexer-digital-electronics **Electronics Tutorials:** https://www.electronics-tutorials.ws/ **All About Circuits:** https://www.electronicsforu.com/videos-slideshows/multiplexer-demultiplexer-basics **YouTube Video Reference:** **Multiplexer and Demultiplexer Explained:** https://m.youtube.com/watch?v=HIeQhZ9Gq5s Homework Use a 4-to-1 multiplexer to implement the following logic function: F(A, B, C) = Σ(1, 2, 5, 6) (where Σ represents the summation of minterms) and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 19	**Course Name:** Computer Architecture & VLSI Design **Topic:** Flip-Flops and Their Types	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic concept and operation of flip-flops. b. Differentiate between various types of flip-flops (SR, D, JK, T). c. Analyze the applications of flip-flops in digital circuits.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever seen a light switch that stays on or off even after you let go of it. - Explain that this persistent memory is a fundamental principle in digital circuits, and flip-flops are the building blocks that enable this behavior. - Introduce the concept of a flip-flop as a circuit that can store a single bit of data. 2. **Development** (30 minutes) **a. SR Flip-Flop:** Explain the basic structure and operation of an SR flip-flop. Discuss the set, reset, and hold states. Highlight the importance of avoiding the forbidden state. **b. D Flip-Flop:** Introduce the D flip-flop as a clocked version of the SR flip-flop. Explain the relationship between the input D and the output Q. Discuss the positive and negative edge-triggered versions. **c. JK Flip-Flop:** Present the JK flip-flop as a more versatile version with both set and reset inputs. Explain the toggle operation when J and K are both high. Discuss the applications of JK flip-flops in sequential circuits. **d. T Flip-Flop:** Introduce the T flip-flop as a simplified version of the JK flip-flop. Explain the toggle operation based on the input T. Discuss the applications of T flip-flops in frequency division circuits. 1. Exercise (5 minutes) – - What is the main function of a flip-flop in digital circuits? - How does a D flip-flop differ from an SR flip-flop? - What happens to a JK flip-flop when both J and K are high. Use Nearpod to collect responses and discuss the answers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Electronics Tutorials:** https://www.electronics-tutorials.ws/ **All About Circuits:** https://www.allaboutcircuits.com/ **Tutorials Point:** https://www.tutorialspoint.com/ **YouTube Video Reference:** **Flip Flops Explained:** https://m.youtube.com/watch?v=Hi7rK0hZnfc Homework Compare and contrast the SR, D, JK, and T flip-flops in terms of their input conditions and state transitions. Discuss scenarios where each type of flip-flop would be preferred over the others and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 20	**Course Name:** Computer Architecture & VLSI Design **Topic:** Level Clocking and Edge Triggered Clocking	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concepts of level clocking and edge-triggered clocking. b. Differentiate between level-sensitive and edge-sensitive flip-flops. c. Analyze the advantages and disadvantages of each clocking scheme.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the concept of synchronous sequential circuits and the importance of clock signals. - Introduce the two primary methods of clocking: level clocking and edge-triggered clocking. 2. **Development** (30 minutes) **a. Level Clocking:** Define level clocking and explain how the data is sampled and stored during a clock pulse. Discuss the types of level-sensitive flip-flops (SR, JK, D, T). Analyze the potential hazards of level clocking (hazards, glitches). **b. Edge-Triggered Clocking:** Define edge-triggered clocking and explain how data is sampled and stored at the rising or falling edge of a clock pulse. Discuss the types of edge-triggered flip-flops (positive-edge and negative-edge triggered). Analyze the advantages of edge-triggered clocking (reduced hazards, simpler design). **c. Comparison:** Compare and contrast level clocking and edge-triggered clocking in terms of speed, simplicity, and susceptibility to hazards. Discuss the trade-offs between the two approaches. 1. Exercise (5 minutes) – - What is the difference between a level-sensitive flip-flop and an edge-triggered flip-flop? - Explain the concept of a hazard in digital circuits. How can hazards be reduced in level-clocked circuits? - Which clocking scheme is generally preferred for high-speed digital systems, and why? Use Nearpod to collect responses and discuss the answers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Edge Triggering and Level Triggering - GeeksforGeeks** **Memory Elements** **Tutorials Point: Edge-Triggered Flip-Flop** **YouTube Video Reference:** **Computer Organization and Architecture A Pedagogical Aspect** **Computer Organization and Architecture NPTEL IITG** **COA [Module 01 - Lecture 02]: Digital Logic Building Blocks - YouTube** Homework Compare and contrast level clocking and edge-triggered clocking in terms of performance and complexity. Which is generally preferred in modern digital systems and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 21	**Course Name:** Computer Architecture & VLSI Design **Topic:** Registers and Their Types	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. understand the concept of registers in computer architecture. b. differentiate between various types of registers based on their functions and uses. c. appreciate the significance of registers in the efficient execution of computer programs.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever heard of a register. - Explain that registers are small, high-speed storage units within the CPU that hold data temporarily during program execution. - Highlight the importance of registers in speeding up computations and improving overall system performance. 2. **Development** (30 minutes) **a. Types of Registers:** **General-purpose registers:** Used for holding data during arithmetic operations, logical operations, and data transfers. **Special-purpose registers:** Designed for specific tasks, such as instruction pointers, program counters, and status registers. **Floating-point registers:** Store and manipulate floating-point numbers. **b. Register Organization:** Discuss how registers are typically organized within the CPU, including the number of registers available and their naming conventions. Explain the concept of register banks and their role in efficient memory access. **c. Register Usage:** Provide examples of how registers are used in different types of instructions, such as load, store, arithmetic, and logical operations. Illustrate the benefits of using registers for temporary data storage and reducing memory access time. 1. Exercise (5 minutes) – - What is the main purpose of general-purpose registers? - Name two types of special-purpose registers. - How do floating-point registers differ from general-purpose registers?

	Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://www.tutorialspoint.com/computer_logical_organization/index.htm https://www.geeksforgeeks.org/what-is-register-digital-electronics/ **https://www.tutorialspoint.com/what-are-computer-registers-in-computer-architecture** YouTube Video Reference: YouTube Homework With advancements in technology, how might the role of registers evolve in future computer architectures and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 22	**Course Name:** Computer Architecture & VLSI Design **Topic:** Bi-Directional Registers	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of bi-directional registers. b. Differentiate between bi-directional and uni-directional registers. c. Explore the applications of bi-directional registers in computer systems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Begin by discussing the concept of registers in computer architecture. - Explain how registers are used to store temporary data during computations. - Introduce the idea of data flow within a register. 2. **Development** (30 minutes) **a. Uni-directional Registers:** Define uni-directional registers. Explain their limitations in terms of data flow. Provide examples of uni-directional registers (e.g., shift registers). **b. Bi-directional Registers:** Define bi-directional registers. Explain how they can transfer data in both directions. Discuss the advantages of bi-directional registers over uni-directional registers. **c. Applications of Bi-directional Registers:** Explore real-world applications of bi-directional registers. Discuss their use in various components of computer systems (e.g., ALU, control unit). Provide examples of specific bi-directional register-based architectures. 1. Exercise (5 minutes) – - What is the difference between a uni-directional and a bi-directional register? - Name two applications of bi-directional registers in computer architecture. - How can bi-directional registers improve the efficiency of data transfer in a computer system? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation

	from students on these. 2. Suggested Reading https://www.tutorialspoint.com/ Shift Registers in Digital Logic - GeeksforGeeks YouTube Video Reference: **https://www.youtube.com/embed/zoEeQgQkPLA** Homework Compare bi-directional registers with parallel-in parallel-out (PIPO) registers in terms of functionality and application and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 23	**Course Name:** Computer Architecture & VLSI Design **Topic:** Memories and Bus Structure: Basic Memory Cell, Memory Hierarchy	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of memory and its different types. b. Learn the structure and function of a basic memory cell. c. Explore the concept of memory hierarchy and its importance in computer systems.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever forgotten something important. Relate this to the concept of memory in computers. - Explain that memory is essential for storing data and instructions in a computer system. - Briefly introduce the different types of memory (primary and secondary). 2. **Development** (30 minutes) **a. Basic Memory Cell:** Explain the structure of a basic memory cell, including its components (address, data, and control lines). Discuss how data is stored and retrieved from a memory cell. **b. Memory Hierarchy:** Define memory hierarchy and its levels (registers, cache, main memory, secondary storage). Explain the advantages and disadvantages of each level in terms of speed, capacity, and cost. Discuss how data is transferred between different levels of the memory hierarchy. **c. Memory Operations:** Explain the basic memory operations (read and write). Demonstrate how data is read from and written to memory. 1. Exercise (5 minutes) – - What are the main components of a basic memory cell? - How does memory hierarchy improve the performance of a computer system? - What is the difference between primary and secondary memory? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

	2. Suggested Reading **Computer Organization and Architecture:** https://vardhaman.org/wp-content/uploads/2021/03/CO.pdf **Memory Hierarchy:** https://www.geeksforgeeks.org/memory-hierarchy-design-and-its-characteristics/ **YouTube Video Reference:** **Basic Memory Cell Explanation:** https://www.youtube.com/watch?v=pDHBOvMTnoY **Memory Hierarchy Tutorial:** https://www.youtube.com/watch?v=_kZY4orPQW0 Homework Discuss a real-world application (e.g., gaming, video editing) and how the choice of memory types and hierarchy impacts performance and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 24	**Course Name:** Computer Architecture & VLSI Design **Topic:** Characteristics, Memory Types, and Accessing Techniques	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. understand the fundamental characteristics of memory. b. differentiate between various memory types. c. learn about different memory accessing techniques.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - what they think memory is and its importance in computers. - Discuss the concept of storage and its role in computer systems. - Introduce the idea that memory is a crucial component for storing data temporarily. 2. **Development** (30 minutes) **a. Characteristics of Memory:** Volatility: Differentiate between volatile and non-volatile memory. Speed: Explain the concept of access time and how it affects memory performance. Capacity: Discuss the different units of measurement for memory capacity (bytes, kilobytes, megabytes, etc.). Cost: Compare the cost per bit of different memory types. **b. Memory Types:** RAM (Random Access Memory): Explain its characteristics, types (DRAM, SRAM), and usage. ROM (Read-Only Memory): Discuss its characteristics, types (PROM, EPROM, EEPROM), and usage. Cache Memory: Explain its purpose, levels (L1, L2, L3), and how it improves performance. Secondary Storage: Briefly introduce devices like hard drives, SSDs, and optical drives as external storage. **c. Accessing Techniques:** Sequential Access: Explain how data is accessed in a sequential manner (e.g., tapes). Direct Access: Discuss how data is accessed directly using a specific address (e.g., hard drives). Random Access: Explain how data can be accessed in any order (e.g., RAM). 1. Exercise (5 minutes) – - What is the difference between volatile and non-volatile memory?

	- Name three types of RAM. - How does cache memory improve computer performance? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **https://www.oempcworld.com/support/glossary.html** **https://www.tutorialspoint.com/computer_concepts/computer_concepts_computer_memory.htm** **YouTube Video Reference:** **https://www.youtube.com/embed/_3SpMwyyFTw** Homework Speculate on the future of memory technologies and accessing techniques. How might advancements change computing in the next decade and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 25	**Course Name:** Computer Architecture & VLSI Design **Topic:** Static and Dynamic Memory, Cache Memory	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of memory allocation in programming. b. Differentiate between static and dynamic memory allocation. c. Learn about cache memory and its role in improving system performance.
Teaching Aids (if any)	a. Interactive Whiteboards b. Examples and diagrams
Teaching Development	1. **Introduction** (5 minutes) - Ask question - the importance of memory in computer systems. - Explain how memory is used to store data and instructions during program execution. - Introduce the concept of memory allocation. 2. **Development** (30 minutes) **a. Static Memory Allocation:** Define static memory allocation. Explain how memory is allocated at compile time. Discuss the advantages and disadvantages of static memory allocation. Provide examples of static variables and arrays. **b. Dynamic Memory Allocation:** Define dynamic memory allocation. Explain how memory is allocated at runtime using functions like malloc, calloc, and realloc. Discuss the advantages and disadvantages of dynamic memory allocation. Provide examples of dynamic memory allocation in C or C++. **c. Cache Memory:** Explain the concept of cache memory. Discuss the different levels of cache (L1, L2, L3). Explain how cache memory improves system performance. Discuss cache coherence and write-back policies. 1. Exercise (5 minutes) – - What is the difference between static and dynamic memory allocation? - Explain the concept of cache memory and its role in improving system performance. - When would you use static memory allocation and when would you use dynamic memory allocation?

	Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Static Memory Allocation:** https://www.tutorialspoint.com/what-is-dynamic-memory-allocation-in-c **Dynamic Memory Allocation:** https://www.tutorialspoint.com/what-is-dynamic-memory-allocation-in-c **Cache Memory:** https://www.tutorialspoint.com/what-is-cache-memory-in-computer-architecture **YouTube Video Reference:** https://www.youtube.com/embed/_3SpMwyyFTw Homework Speculate on the future of memory technologies and accessing techniques. How might advancements change computing in the next decade and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 26	**Course Name:** Computer Architecture & VLSI Design **Topic:** Memory Address Map to CPU	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of memory address space. b. Learn how memory addresses are assigned to different components of a computer system. c. Grasp the role of the Memory Management Unit (MMU) in mapping virtual addresses to physical addresses.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Begin with a real-world analogy: A city map. Explain how addresses are used to locate specific locations in a city. - Introduce the concept of memory as a vast storage space. - Highlight the need for a systematic way to access and manage this memory space. 2. **Development** (30 minutes) **a. Memory Address Space:** Explain the binary representation of memory addresses. Discuss the concept of byte-addressable memory. Illustrate how memory is divided into smaller units like bytes, words, and double words. **b. Memory Mapping:** Describe the process of assigning memory addresses to different components like RAM, ROM, I/O devices, and the operating system. Explain the role of the Memory Management Unit (MMU) in translating virtual addresses to physical addresses. Discuss the concept of virtual memory and its benefits. **c. Memory Hierarchy:** Introduce the concept of a memory hierarchy, including registers, cache, main memory, and secondary storage. Explain how data is transferred between different levels of the memory hierarchy. Discuss the impact of memory hierarchy on system performance. 1. Exercise (5 minutes) – - What is the purpose of a memory address? - How does the MMU help in efficient memory management? - What is the difference between physical memory and virtual memory?

	Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **How to Address Mapping using Pages in Computer Architecture?** **https://www.geeksforgeeks.org/memory-management-in-operating-system/** **YouTube Video Reference:** **https://www.youtube.com/watch?v=jSC49w3M05E** Homework What is a memory address map, and why is it important for the CPU and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 27	**Course Name:** Computer Architecture & VLSI Design **Topic:** Bus Structure	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of a bus in computer architecture. b. Learn the different types of buses and their functions. c. Analyze the advantages and disadvantages of bus architecture.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - if they have ever seen a highway. Explain how a highway is a one-way or two-way road that allows vehicles to move from one place to another. - Relate this concept to a computer's bus, which is a pathway that allows data to move between different components. - Introduce the basic components of a computer system (CPU, memory, input/output devices) and how they communicate with each other. - **Development** (30 minutes) **a. Types of Buses:** **Data Bus:** Transfers data between components. **Address Bus**: Carries memory addresses. **Control Bus:** Transmits control signals to coordinate data transfer. **b. Bus Width:** Explain how bus width affects data transfer rate. Discuss the impact of wider buses on performance. **c. Bus Arbitration:** Introduce the concept of bus arbitration. Explain how devices request and gain control of the bus. 1. Exercise (5 minutes) – - What are the three main types of buses in a computer system? - How does bus width affect the performance of a computer? - What is the purpose of a control bus? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **How to Address Mapping using Pages in Computer Architecture?** **https://www.geeksforgeeks.org/memory-management-in-operating-system/**

	YouTube Video Reference: **https://www.youtube.com/watch?v=jSC49w3M05E** Homework What is a memory address map, and why is it important for the CPU and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 28	**Course Name:** Computer Architecture & VLSI Design **Topic:** Memory-Mapped and I/O-Mapped Techniques	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of memory-mapped and I/O-mapped I/O techniques. b. Differentiate between the two techniques in terms of their implementation and performance characteristics. c. Analyze the advantages and disadvantages of each technique.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly introduce the concept of input/output (I/O) operations in computer systems. - Explain the need for efficient I/O techniques to interact with external devices. - Highlight the two primary methods for I/O: memory-mapped and I/O-mapped. 2. **Development** (30 minutes) **a. Memory-Mapped I/O:** Explain the concept of treating I/O devices as memory locations. Discuss the use of memory-mapped I/O instructions to access and control devices. Highlight the advantages of memory-mapped I/O, such as simplified programming and efficient data transfer. **b. I/O-Mapped I/O:** Introduce the concept of dedicated I/O instructions to communicate with devices. Explain the use of I/O ports to address and access devices. Discuss the advantages of I/O-mapped I/O, such as dedicated hardware support and potential performance benefits. **c. Comparison and Trade-offs:** Compare and contrast the two techniques in terms of hardware complexity, software complexity, and performance. Discuss the factors to consider when choosing between the two techniques. 1. Exercise (5 minutes) – - What is the key difference between memory-mapped and I/O-mapped I/O techniques? - Explain the concept of an I/O port. - What are the advantages and disadvantages of memory-mapped I/O?

	Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **https://www.geeksforgeeks.org/difference-between-memory-mapped-io-and-io-mapped-io-with-reference-to-8085-microprocessor/** **https://www.geeksforgeeks.org/difference-between-memory-mapped-io-and-io-mapped-io-with-reference-to-8085-microprocessor/** **YouTube Video:** **https://m.youtube.com/watch?v=vxotcZdEns4** Homework Provide examples of computer architectures that utilize Memory-Mapped I/O and those that use I/O-Mapped I/O. What are the reasons for their design choices and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 29	**Course Name:** Computer Architecture & VLSI Design **Topic:** Modes of I/O transfers	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of Input/Output (I/O) operations in computer systems. b. Learn about the different modes of I/O data transfer: Programmed I/O, Interrupt-Driven I/O, and Direct Memory Access (DMA). c. Analyze the advantages and disadvantages of each mode and their suitable applications.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - explaining the importance of I/O operations in computer systems. - Discuss how computers interact with external devices like keyboards, monitors, hard drives, etc. - Introduce the concept of data transfer between the CPU and I/O devices, highlighting the need for efficient mechanisms. 2. **Development** (30 minutes) **a. Programmed I/O:** Explain the basic concept: CPU directly controls I/O operations. Discuss the process: CPU polls the device status, transfers data, and waits for the next operation. Highlight the drawbacks: CPU-bound, inefficient for large data transfers. **b. Interrupt-Driven I/O:** Introduce the concept: I/O devices generate interrupts to signal the CPU. Explain the process: CPU handles the interrupt, services the I/O request, and returns to its original task. Discuss the advantages: Improved efficiency compared to programmed I/O. **c. Direct Memory Access (DMA):** Explain the concept: DMA controller handles data transfer directly between I/O devices and memory. Discuss the process: DMA controller takes control of the bus, transfers data, and signals the CPU upon completion. Highlight the advantages: High-speed data transfer, relieves the CPU from I/O operations. 1. Exercise (5 minutes) – - What is the primary difference between Programmed I/O and Interrupt-Driven I/O?

	- In which I/O mode does the CPU have the least involvement in data transfer? - What is the role of a DMA controller in a computer system? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **https://www.techtarget.com/whatis/definition/input-output-I-O** **https://www.hostreview.com/blog/230630-different-types-of-data-transfers** **YouTube Video:** **https://www.youtube.com/watch?v=VZxen9Ekuog** Homework For each mode of I/O transfer (programmed, interrupt-driven, DMA), provide an example of a real-world application or device that typically uses that mode. Justify your choices and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 30	**Course Name:** Computer Architecture & VLSI Design **Topic:** Instruction & Interrupt Life Cycle	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concept of the instruction cycle. b. Learn the different phases of the instruction cycle. c. Grasp the concept of interrupts and their role in system efficiency. d. Understand the interrupt life cycle and its phases. e. Apply the knowledge of instruction and interrupt cycles to analyze system performance.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Begin with a brief overview of computer architecture, highlighting the central processing unit (CPU) as the brain of the computer. - Introduce the concept of instructions as the commands that the CPU executes to perform tasks. - Explain the need for a systematic process to execute these instructions, leading to the introduction of the instruction cycle. - Briefly touch upon the concept of interrupts as events that can disrupt the normal flow of the instruction cycle. 2. **Development** (30 minutes) **a. Instruction Cycle:** Explain the four main phases of the instruction cycle: Fetch, Decode, Execute, and Write-back. Discuss the role of the program counter (PC) in keeping track of the next instruction to be fetched. Elaborate on the control unit's role in controlling the execution of each phase. Use diagrams and analogies to illustrate the concept. **b. Interrupt Life Cycle:** Define interrupts as signals that inform the CPU about an event that requires immediate attention. Explain the steps involved in handling an interrupt: Interrupt request Interrupt acknowledgment Saving the current state of the CPU Transferring control to the interrupt service routine (ISR) Executing the ISR Restoring the CPU's state Returning to the interrupted program c. Discuss the importance of interrupts in handling I/O operations,

	system timers, and other asynchronous events. 1. Exercise (5 minutes) – - What are the four main phases of the instruction cycle? - What is the role of the interrupt service routine (ISR)? - How does the CPU handle multiple interrupts simultaneously? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Textbooks:** Computer Organization and Design: The Hardware/Software Interface by Patterson and Hennessy Computer Architecture: A Quantitative Approach by Hennessy and Patterson **Online Resources:** https://www.geeksforgeeks.org/different-instruction-cycles/ https://www.tutorialspoint.com/what-is-instruction-cycle-in-computer-architecture **YouTube Video Reference:** https://m.youtube.com/watch?v=UN6EsWTTofA Homework Explain the differences between hardware interrupts, software interrupts, and timer interrupts. Provide examples of each type. Compare and contrast the different types of instructions (data processing, control flow, data movement). How does each type affect the instruction life cycle and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 31	**Course Name:** Computer Architecture & VLSI Design **Topic:** VHDL components and tools	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of VHDL components and their role in modular design. b. Learn how to create and instantiate components in VHDL code. c. Explore popular VHDL synthesis and simulation tools. d. Gain hands-on experience with VHDL design flow.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly introduce VHDL as a hardware description language (HDL) used to design digital circuits. - Explain the importance of modular design in complex digital systems. - Introduce the concept of VHDL components as building blocks for modular design. - Highlight the role of VHDL tools in the design and verification process. 2. **Development** (30 minutes) **a. VHDL Components:** Define components as self-contained units of VHDL code with specific inputs and outputs. Explain the syntax for component declaration and instantiation. Demonstrate how to use components to create hierarchical designs. **b. VHDL Design Flow:** Outline the typical VHDL design flow: Design entry (textual or graphical) Synthesis (translation of VHDL code into hardware) Simulation (verification of design behavior) Implementation (physical design and layout) Introduce popular VHDL synthesis and simulation tools like Xilinx Vivado, ModelSim, and VHDL simulators. **c. Hands-on Exercise (if time permits):** Guide students through a simple VHDL design using components: Create a component for a 2-input AND gate. Instantiate multiple AND gates to create a 4-input AND function. Simulate the design to verify its functionality. 1. Exercise (5 minutes) –

	- What is a VHDL component? - Name two popular VHDL synthesis and simulation tools. - What is the purpose of the component declaration in VHDL? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Textbooks:** VHDL Programming by J. Bhasker Digital Design with VHDL by M. Morris Mano **Online Resources:** Xilinx Documentation: https://www.amd.com/en/products/adaptive-socs-and-fpgas/technologies.html ModelSim Documentation: https://eda.sw.siemens.com/en-US/ic/modelsim/ VHDL Tutorial: https://nandland.com/introduction-to-vhdl-for-beginners-with-code-examples/ **YouTube Video Reference:** https://www.youtube.com/watch?v=BDq8-QDXmek Homework Describe the role of a VHDL simulation tool in the design process. Mention at least two popular VHDL simulation tools and their key features and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 32	**Course Name:** Computer Architecture & VLSI Design **Topic:** Introduction to VHDL: Need and Importance	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic concepts of Hardware Description Languages (HDLs). b. Learn about the history and evolution of VHDL. c. Grasp the significance of VHDL in modern digital design. d. Identify the key applications of VHDL in industry. e. Develop a basic understanding of VHDL syntax and structure.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - introducing the concept of digital design and the challenges associated with designing complex digital circuits. - Explain the need for a systematic and efficient method to describe hardware behavior. - Introduce Hardware Description Languages (HDLs) as a solution to these challenges. - Highlight the specific role of VHDL as a powerful HDL for designing digital systems.. 2. **Development** (30 minutes) **a. What is VHDL?** Define VHDL as a standardized language for describing digital systems. Explain the key features of VHDL, such as its ability to model both behavioral and structural aspects of hardware. Discuss the hierarchical design methodology supported by VHDL. **b. Why VHDL?** Highlight the importance of VHDL in the design and verification of complex digital systems. Discuss the advantages of using VHDL, including increased design productivity, reduced time-to-market, and improved design quality. Explain the role of VHDL in various industries, such as electronics, telecommunications, and aerospace. **c. Basic VHDL Syntax:** Introduce the basic structure of a VHDL design unit, including entity and architecture declarations. Explain the concept of ports and signals. Provide a simple example of a VHDL code to demonstrate the basic syntax. 1. Exercise (5 minutes) – - What is a Hardware Description Language (HDL)?

	- Why is VHDL a popular choice for digital design? - What are the two main components of a VHDL design unit? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **Textbooks:** Digital Design Principles and Practices by John F. Wakerly VHDL Programming by Douglas L. Perry **Online Resources:** Xilinx University Program: https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2015x/VHDL/docs-pdf/Vivado_Tutorial.pdf FPGA4U: https://fpgatutorial.com/vhdl/ **YouTube Video Reference:** https://www.youtube.com/watch?v=BDq8-QDXmek Homework Describe the role of a VHDL simulation tool in the design process. Mention at least two popular VHDL simulation tools and their key features and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 33	**Course Name:** Computer Architecture & VLSI Design **Topic:** Characteristics, Basic Components of VHDL	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of VHDL. b. Learn the basic syntax and structure of VHDL code. c. Identify the key components of a VHDL design unit.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly introduce VHDL as a Hardware Description Language (HDL) used to design digital circuits. - Highlight its importance in the field of digital electronics and VLSI design. - Explain the advantages of using VHDL over traditional design methods. 2. **Development** (30 minutes) **a. Characteristics of VHDL:** Text-based language for describing digital circuits. Hierarchical design methodology. Concurrent nature of operations. Strong typing system. Support for various design abstraction levels (behavioral, RTL, gate-level). **b. Basic Components of VHDL:** Entity: Declares the interface of a design unit. Architecture: Defines the implementation of the design unit. Signal: Used to represent data paths and control signals. Process: A sequential block of statements. Concurrent Statement: A statement executed concurrently with others. Data Types: Standard data types (bit, std_logic, integer, etc.) and user-defined types. 1. Exercise (5 minutes) – - What is the primary purpose of an entity declaration in VHDL? - Explain the difference between a signal and a variable in VHDL. - What is the significance of concurrent statements in VHDL? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

	2. Suggested Reading **VHDL Tutorial:** **https://www.eecs.umich.edu/courses/doing_dsp/handout/vhdl-tutorial.pdf** **VHDL Language Reference Manual: https://faculty-web.msoe.edu/johnsontimoj/Common/FILES/VHDL_2008.pdf** **YouTube Video Reference:** **VHDL Tutorial for Beginners:** **https://www.youtube.com/watch?v=BDq8-QDXmek** Homework What is the significance of VHDL being a strongly typed language? Provide examples of how strong typing affects design. and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 34	**Course Name:** Computer Architecture & VLSI Design **Topic:** Simple VHDL Program	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic structure of a VHDL program. b. Learn to write simple VHDL code for basic logic gates. c. Simulate and analyze the designed VHDL code.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Brief overview of VHDL as a Hardware Description Language (HDL). - Explain its use in designing digital circuits. - Introduce the basic structure of a VHDL program, including entity and architecture. 2. **Development** (30 minutes) **a. Entity Declaration:** Define the input and output ports of the module. Use the entity keyword to declare the module name and port list. **b. Architecture Body:** Describe the functional behavior of the module. Use the architecture keyword to specify the architecture name and behavioral description. Utilize VHDL operators and signal assignments to implement logic functions. **c. Simple VHDL Example: 2-input AND Gate** Write a VHDL code for a 2-input AND gate. Explain the code line by line. Demonstrate the simulation of the code using a VHDL simulator. 1. Exercise (5 minutes) – - What are the two main parts of a VHDL module? - What is the purpose of the entity declaration? - Write a VHDL code for a 2-input OR gate. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading VHDL Tutorial: https://nandland.com/introduction-to-vhdl-for-beginners-with-code-examples/ VHDL Language Reference Manual: https://faculty-web.msoe.edu/johnsontimoj/Common/FILES/VHDL_2008.pdf

	YouTube Video Reference: VHDL Tutorial for Beginners: https://www.youtube.com/watch?v=BDq8-QDXmek Homework Design a VHDL program for a full adder. The program should have three inputs (A, B, Cin) and two outputs (Sum, Cout). Create a testbench to demonstrate its operation. Write a VHDL program for a 4-to-1 multiplexer. Include the logic for selecting between four input signals based on a 2-bit select line. Test the multiplexer using a testbench and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 35	**Course Name:** Computer Architecture & VLSI Design **Topic:** GHDL VHDL Simulator	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. introduce students to the concept of VHDL simulation. b. familiarize students with the GHDL simulator. c. guide students on how to install and set up GHDL on their systems. d. demonstrate basic VHDL simulation using GHDL.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly explain the importance of simulation in digital design. - Introduce VHDL as a hardware description language. - Highlight the role of simulators in verifying the correctness of VHDL designs. - Introduce GHDL as a free and open-source VHDL simulator. 2. **Development** (30 minutes) **a. Installing GHDL:** Provide instructions on how to download and install GHDL based on the operating system (Windows, Linux, macOS). Explain the importance of adding GHDL to the system's PATH variable. **b. Basic VHDL Simulation:** Create a simple VHDL module (e.g., a 2-input AND gate). Write a testbench to stimulate the module. Demonstrate how to compile and simulate the design using GHDL commands. Explain the concept of simulation waveforms and how to interpret them. **c. Advanced Features of GHDL:** Briefly discuss advanced features like waveform viewing, debugging, and code coverage analysis. Encourage students to explore the GHDL documentation for more details. 1. Exercise (5 minutes) – - What is the primary purpose of a VHDL simulator? - Name a free and open-source VHDL simulator. - What is a testbench in VHDL simulation? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading

	GHDL Official Website: https://github.com/gcc-mirror/gcc/blob/master/gcc/config/gnu.h GHDL Documentation: https://www.gnuhealth.org/docs/ VHDL Tutorial: https://nandland.com/introduction-to-vhdl-for-beginners-with-code-examples/ Homework Design a 4-bit up counter in VHDL. Write the VHDL code, compile it using GHDL, create a testbench, and run the simulation. Document the entire process step-by-step and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 36	**Course Name:** Computer Architecture & VLSI Design **Topic:** Xilinx ISE (FPGA Synthesis Toolset)	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic concepts of FPGA design flow. b. Learn how to use Xilinx ISE to create and synthesize FPGA designs. c. Gain hands-on experience with the Xilinx ISE toolset.
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly introduce Field Programmable Gate Arrays (FPGAs) and their advantages over traditional microprocessors. - Explain the role of FPGA synthesis tools in the design process. - Overview of the Xilinx ISE toolset and its primary components.. 2. **Development** (30 minutes) **a. Xilinx ISE Design Flow:** High-level design entry using VHDL or Verilog. Synthesis process: translating the design into a gate-level netlist. Implementation process: mapping the netlist to the target FPGA device. Place and Route: assigning physical locations to logic elements and routing interconnections. **b. Xilinx ISE Tool Usage:** Creating a new project in Xilinx ISE. Adding design source files (VHDL or Verilog). Setting up synthesis and implementation constraints. Running the synthesis and implementation processes. Analyzing design reports and timing constraints. **c. Debugging and Verification:** Using simulation tools to verify the functional correctness of the design. Identifying and fixing design errors. Analyzing timing reports to optimize the design for performance. 1. Exercise (5 minutes) – - What is the primary function of the synthesis tool in the FPGA design flow? - Name the two primary hardware description languages used for FPGA design. - What is the significance of timing constraints in FPGA design? Use Nearpod to collect responses and discuss the answers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Xilinx Official Website: https://www.xilinx.com/ Homework Outline the steps to create a simple LED blinking design in Xilinx ISE. Include HDL code snippets. How would you debug a timing violation error during the implementation phase? Describe the tools and methods you would use. What considerations must be taken into account when targeting different FPGA architectures with Xilinx ISE and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 37	**Course Name:** Computer Architecture & VLSI Design **Topic:** Image Simulation Accelerator (FPGA-based Co-simulation Environment)	**Course No.: MCA-103**

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of hardware-software co-simulation b. Learn about the role of FPGAs in accelerating image processing tasks c. Explore the benefits and challenges of FPGA-based image simulation accelerators d. Gain knowledge of popular tools and techniques used in FPGA-based co-simulation
Teaching Aids (if any)	a. Interactive Whiteboards b. Diagrams and flowcharts
Teaching Development	1. **Introduction** (5 minutes) - Ask question - Briefly introduce the concept of hardware-software co-simulation and its applications in various domains, including image processing. - Highlight the limitations of traditional software-based image processing techniques, particularly for real-time and computationally intensive tasks. - Introduce FPGAs as a powerful hardware platform for accelerating image processing algorithms. 2. **Development** (30 minutes) **a. FPGA-based Co-simulation Environment:** Explain the basic architecture of an FPGA-based co-simulation environment, including the host computer, FPGA board, and communication interfaces. Discuss the role of simulation tools and frameworks in integrating hardware and software components. Highlight the importance of efficient data transfer and synchronization between the host and FPGA. **b. Image Processing Algorithms on FPGAs:** Identify suitable image processing algorithms for FPGA acceleration, such as convolution, filtering, and feature extraction. Discuss the challenges of mapping these algorithms onto FPGA hardware, including resource constraints and timing requirements. Explore techniques for optimizing FPGA implementations, such as pipelining, parallelism, and memory hierarchy optimization. **c. Real-world Applications:** Present real-world examples of FPGA-based image simulation accelerators, such as real-time video processing, medical image analysis, and autonomous vehicle perception. Discuss the performance benefits and cost-effectiveness of FPGA-based solutions compared to traditional software-based

	approaches. 1. Exercise (5 minutes) – - What is the primary advantage of using FPGAs for image processing acceleration? - Explain the concept of hardware-software co-simulation in the context of image processing. - What are some common challenges in mapping image processing algorithms onto FPGAs? Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Xilinx: https://www.amd.com/en/products/adaptive-socs-and-fpgas/technologies.html ntel FPGA: https://www.intel.com/content/www/us/en/products/details/fpga.html MathWorks: https://www.mathworks.com/ OpenCL: https://opencl.org/ Homework Discuss a real-world application where an Image Simulation Accelerator might be utilized. Include specific image processing tasks that could benefit from FPGA implementation. How can one optimize the performance of an ISA when simulating complex image algorithms? Consider factors like resource allocation, data throughput, and latency and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents